Patent Claims

1. Hardware implemented filtering method comprising the steps of
- establishing a representation (DIS) of the derivative of at least a part of a time-quantized input signal (IS), and
- establishing at least one sample of a time- and amplitude-quantized output signal (OS) by performing filtering on the basis of at least a part of a filter representation (IFC1, IFC2, IFC3) and said representation (DIS) of the derivative of at least a part of said input signal (IS).

2. Hardware implemented method of convolving in the time domain an input signal ($x[n]$) with an impulse response ($h[k]$) in order to establish an output signal ($y[n]$), characterised by that said output signal ($y[n]$) is provided at least partly by a convolution in the time domain of a difference signal representation ($x'[n]$) of said input signal ($x[n]$) and a sum representation ($l[k]$) of said impulse response ($h[k]$).

3. Hardware implemented filtering method according to claim 1, whereby said step of establishing at least one sample of a time- and amplitude-quantized output signal (OS) is implemented according to the method of claim 2.

4. Hardware implemented filtering method according to any of the claims 1 to 3, whereby said impulse response is finite.

5. Hardware implemented filtering method according to any of the claims 1 to 4, whereby said time-quantized input signal (IS) comprises in average at least 10, preferably at least 64, and even more preferably at least 128 samples for each input signal value change.

6. Hardware implemented filtering method according to any of the claims 1 to 5, whereby said time-quantized input signal (IS) is a pulse width modulated signal.

7. Hardware implemented filtering method according to any of the claims 1 to 6, whereby said establishing a representation (DIS) of the derivative of at least a part of said time-quantized input signal (IS) comprises the step of establishing a sequence of differences between successive samples of said at least a part of said input signal (IS).

8. Hardware implemented filtering method according to any of the claims 1 to 7, whereby said at least a part of said time-quantized input signal (IS) in respect of its length corresponds to the length of said at least a part of an impulse response.

9. Hardware implemented filtering method according to any of the claims 1 to 8, whereby said representation (DIS) of the derivative of at least a part of said time-quantized input signal (IS) is stored in a differentiated input signal representing array (DA).

10. Hardware implemented filtering method according to any of the claims 1 to 9, whereby said establishing a representation (DIS) of the derivative of at least a part of a time-quantized input signal (IS) comprises the step of indexing corresponding times and directions of amplitude changes of said at least a part of said input signal (IS).

11. Hardware implemented filtering method according to any of the claims 1 to 10, whereby the length of said at least a part of said filter representation (IFC1, IFC2, IFC3) is an integer multiple of the length of a symbol of said at least a part of said time-quantized input signal (IS).

12. Hardware implemented filtering method according to any of the claims 1 to 11, whereby the number of changes within a symbol of said at least a part of said time-quantized input signal is constant.

13. Hardware implemented filtering method according to any of the claims 1 to 12, whereby said times are indexed relative to each other.

14. Hardware implemented filtering method according to any of the claims 1 to 13, whereby said establishing a representation (DIS) of the derivative of at least a part of a time-quantized input signal (IS) comprises the step of storing into a snapshot register (SR) said at least a part of said time-quantized input signal (IS).

15. Hardware implemented filtering method according to any of the claims 1 to 14, whereby said establishing a representation (DIS) of the derivative of at least a part of a time-quantized input signal (IS) comprises the step of querying said snapshot register (RS) regarding input signal changes.

16. Hardware implemented filtering method according to any of the claims 1 to 15, whereby said at least a part of said filter representation (IFC1, IFC2, IFC3) is a sum representation of at least a part of an impulse response.

17. Hardware implemented filtering method according to any of the claims 1 to 16, whereby said at least a part of said filter representation (IFC1, IFC2, IFC3) is predetermined.

18. Hardware implemented filtering method according to any of the claims 1 to 17, whereby said at least a part of said filter representation (IFC1, IFC2, IFC3) is implemented by means of at least one filter coefficient, more preferably at least 128 filter coefficients and even more preferably at least 384 filter coefficients.

19. Hardware implemented filtering method according to any of the claims 1 to 18, whereby said at least a part of said filter representation (IFC1, IFC2, IFC3) is implemented by means of at least one model, preferably represented by at least one polynomial.

20. Hardware implemented filtering method according to any of the claims 1 to 19, whereby said implementation of said at least a part of said filter representation (IFC1, IFC2, IFC3) is adapted to utilize any symmetry of said filter representation.

21. Hardware implemented filtering method according to any of the claims 1 to 20, whereby said at least a part of said filter representation (IFC1, IFC2, IFC3) is user-adjustable.

22. Hardware implemented filtering method according to any of the claims 1 to 21, whereby said performing filtering comprises convolving said at least a part of said filter representation (IFC1, IFC2, IFC3) with said representation (DIS) of the derivative of at least a part of said time-quantized input signal (IS).

23. Hardware implemented filtering method according to any of the claims 1 to 22, whereby said performing filtering further comprises for each of said at least one sample of a time- and amplitude-quantized output signal (OS) adding the result of multiplying an initial value (IV) of said at least a part of said time-quantized input signal (IS) with a value of said at least a part of said filter representation (IFC1, IFC2, IFC3).

24. Hardware implemented filtering method according to any of the claims 1 to 23, whereby said performing filtering further comprises adding, for each of said at least one sample of a time- and amplitude-quantized output signal (OS), an initial value (IV) of said at least a part of said time-quantized input signal (IS).

25. Hardware implemented filtering method according to any of the claims 1 to 24, whereby said performing filtering comprises exercising the expression $y[n] = \sum_{k=0}^{N-2}\left(l[k]\cdot x'[n-k]\right) + l[N-1]\cdot x[n-(N-1)]$, where $y[n]$ represents said at least one sample of a time- and amplitude-quantized output signal (OS), $x[n]$ represents said at least a part of said time-quantized input signal (IS), $x'[n]$ represents said representation (DIS) of the derivative of $x[n]$, $l[k]$ represents said at least a part of said filter representation (IFC1, IFC2, IFC3), and N represents the length of $l[k]$.

26. Hardware implemented filtering method according to any of the claims 1 to 25, whereby said performing filtering further comprises performing conventional filtering.

27. Hardware implemented filtering method according to any of the claims 1 to 26, whereby the sample rate of said time- and amplitude-quantized output signal (OS) is different from the sample rate of said time-quantized input signal (IS).

28. Hardware implemented filtering method according to any of the claims 1 to 27, whereby the sample rate of said time- and amplitude-quantized output signal (OS) corresponds to the symbol rate of said time-quantized input signal (IS).

29. Hardware implemented filtering method according to any of the claims 1 to 28, whereby said convolving said at least a part of said filter representation (IFC1, IFC2, IFC3) with said representation (DIS) of the derivative of at least a part of said time-quantized input signal (IS) is performed for only some of the samples of said time-quantized input signal (IS), preferably for only every 128^{th} sample.

30. Hardware implemented filtering method according to any of the claims 1 to 29, whereby said filter representation (IFC1, IFC2, IFC3) comprises a sum representation of a low-pass filter.

31. Hardware implemented filtering method according to any of the claims 1 to 30, whereby said method is exercised in real time.

32. Hardware implemented filtering method according to any of the claims 1 to 31, whereby said at least a part of a filter representation (IFC1, IFC2, IFC3) represents at least a part of an impulse response.

33. Hardware implemented filtering method according to any of the claims 1 to 32, whereby said at least a part of a filter representation (IFC1, IFC2, IFC3) represents the derivative of at least a part of an impulse response.

34. Hardware implemented filtering method according to any of the claims 1 to 33 further comprising the step of

- integrating at least once said time- and amplitude-quantized output signal (OS).

35. Hardware implemented decimation method for decimating a time-quantized input signal (IS) comprising the steps of

- dividing said time-quantized input signal (IS) into intervals,
- for each of said intervals establishing a sample of a time- and amplitude-quantized output signal (OS) according to any of the claims 1 to 34.

36. Fast filtering means (FFM) comprising

- differentiation means (DM) for establishing a representation (DIS) of the derivative of at least a part of a time-quantized input signal (IS), and
- filtering means (FM) for establishing at least one sample of an output signal (OS) by performing filtering on the basis of at least a part of a filter representation (IFC1, IFC2, IFC3) and said representation (DIS) of the derivative of at least a part of said input signal (IS).

37. Fast filtering means (FFM) according to claim 36 implementing the hardware implemented filtering method according to any of the claims 1 to 34 or the hardware implemented decimation method according to claim 35.